REMARKS

The applicant appreciates the examiner's cooperation in forwarding by fax the text of the Detailed Action document because the wrong Detailed Action (one for SN 09/353,998) was originally attached to the cover page of the official communication. Because of the prompt response of the examiner, no resetting of the time for response is anticipated. Additional discussion was had regarding the nature of the background of the invention in telephonic communication, but no agreement was reached.

The applicant has modified claims 1 and 17 responsive to the challenges in understanding identified in the official communication for these claims. It is believed that they are now more clear, but have not been altered in scope. It is also believed that after review of the discussion regarding the concerns raised by the examiner under the umbrella of section 112 rejections, a greater understanding of the nature of the invention will be apparent.

Toward that greater understanding, the applicant refers the examiner to the following link, which may provide some useful background, although it too is not believed to provide any negative to patentability of the presently considered claims. **http://csdl.computer.org/comp/trans/td/1993/02/l0131abs.htm**

Discussion of the Section 112 rejection.

1. This invention relies upon processor associated switching queues (ex: Claim 1, line 9) and uses affinity assignments of processors to those queues (claim 1 lines 11 and 12), along with additional rules to allocate run time for the processors to the tasks (lines 13-22). The cited art does not describe any of these limitations.

2. The particulars of how the invention addresses these limitations while the cited art does not will become clear in going through the section 112 issues raised in the official communication. It is believed that the section 112 rejections are fully

met by the discussion below and the two amendments to claims 1 and 17, highlighted in this discussion.

a. Regarding lines 9-10 of claim 1, the first section 112 concern, the claim says that we are limited to a "processor associated switching queue." This limitation indicates that there is at least a switching queue associated with a processor, and that this switching queue is being used in this method. See page 6, line 21 to page 7 line 12 for an example support of this limitation in the specification. See also the first paragraph on page 8 indicating that this queue can be unique. Since we are talking about an Operating System (or OS as it may be referred to), it is clear from this description that the OS maintains the association between the switching queue and the processor. In the page 7 paragraph referred to, a preferred embodiment is described as having six bits for affining a task to a switching queue or SQ. This six bit structure can be called a control data structure for the task, as one of ordinary skill in this art may call it. In the preferred embodiment it is referred to as a header.

b. Regarding line 11, it is referring to the same new task. The claim is modified to clarify this point. No change of scope is made by this clarification.

c. Regarding lines 15-16, overuse of pronouns is eliminated by amendment to clarify this language. No change of scope is made by this clarification.

d. Regarding line 18, the second level switching queue is variable. Note the paragraph bridging pages 8 and 9 for detailed explanation. The language of the claim seems appropriate to describe this. No "first level" is used so that there is no confusion between a second level for one processor being a first level for another. Each processor has a SQ associated with it, and has the opportunity to use other switching queues as described in the specification. Thus, these other switching queues when used are second level switching queues for the processor using them instead of its processor associated switching queue. A second level switching queue could alternatively be an SQ used for a group of processors. If one wanted

to call the processor associated switching queue a first level switching queue, one could perhaps do that, but it is believed that the current claim language is more clear, and eliminates the confusion that may result from such a change in language.

e. Claim 17's "problems" should satisfactorily resolve with reference to the clarifying statements in a-d above and the clarifying amendment made also to claim 17, identical to the amendment made to claim 1.

f. The questions about claim 18 and by implication claims 22 and 23 are not fully understood, and if continued, further explanation is required. In Fig. 4 the number of processors and actual switching queues is the same. Using the word "substantial" in this circumstance allows the inventors to capture with their claims any infringers that may choose to disable a processor, or on the other hand, say that a dedicated maintenance processor which has no switching queue excuses them from infringement because they don't have exactly the same number of SQ's as processors. This is believed to be a common usage in patent claims practice, and one that has been well accepted. Likewise, infringers who would claim that using one SQ for another processor as a second level switching queue is actually using a new SQ, or a copy of it temporarily for that purpose could conceivably survive infringement claims based on such an argument. Systems using groups of processor SQ's as well as the SQ processor paired SQ's would also be avoiding infringement despite using the inventive concepts. Accordingly, it is very reasonable to use the word substantially to avoid loss of the rights to which this invention is entitled. The word substantially has been found in many patents for similar reasons and should not be objectionable here.

g. Lines 14- 15 of claim 18 do not appear to have the complained of language. Reconsideration or clarification is respectfully requested.

3. Reconsideration and withdrawal of the section 112 rejection is respectfully solicited.

Discussion of rejection under section 103.

The Steely reference is first cited, with a conclusion that it has the same memory structure as is identified in the beginning of claim 1. The applicant respectfully disagrees with this conclusion as being unsupported by the reference. Looking at the cited column 5 language and the Fig. 1 description, it is plain that there are two levels of memory in Steely, a cache for each instruction processor 102, 104, 106, and 108 and for the IOP 130, and there is another level of memory above that in the hierarchy, i.e., shared memory 150. There is a reference to an additional memory the DTAG 160, but this memory is used to capture metadata about the state of cache coherency, not to provide an intermediate level cache memory as is the case in applicant's preferred hardware. DTAG is very clearly not within the hierarchy of memory through which data must pass to go between the instruction processor (or I/O processor) and the main (or shared) memory. In conclusion then, there is no three level memory hierarchy in Steely.

Contrast this organization with the hardware organization which is identified in the claims, where the applicant describes at least three levels of memory hierarchy. With reference to applicant's Fig. 1, there are not just three but four levels of memory hierarchy defined, an FLC, a SLC, a TLC and then the main memory, the MSU. Data moves through the hierarchy to get from one point to another, say from an IP to the MSU, for example, where it has to go through each level of the hierarchy. The Steely reference shows a mere two hierarchical levels of memory in its architecture. We have reviewed each of the cited sections of Steely and not found anything to contradict this finding. Therefore the Steely reference may not be appropriately applied to the claims as it does not appear to show three levels of hierarchical memory structure.

Within the same paragraph 6 of the official communication, Steely is again referenced to suggest that it contains or describes "...a process of assigning a task to the selected switching queue by placing information about an affined switching queue into said new task running a one of the instruction processors based upon

tasks ..., citing column 4 lines 20 to 30 (page 4 of the official communication). Firstly, the cited language appears to be discussing how the operation of the previous sentence works. The previous sentence says that ... *"Each processor preferably has a private cache for storing data and changes to the data as a result of the memory reference operations are reflected among the entities via the transmission of probe signals through the probe queues in accordance with a conventional cache coherence protocol. Associated with each switch is an ordering point that serializes memory reference operations, including the MB* [Memory Barrier] *operation, issued to the system and generates the MB-Ack in response to the MB operation."*. Thus, if what Steely is saying here is germane to any task, it is to the task of serializing memory reference operations including MB (Memory Barrier (see lines 29 et seq of column 3)) operations.

While it should be clear that the applicant is not talking about a single memory read or write as a "task", reference to the description of Fig. 2 in the first paragraph of page 6 of the application should make that very clear. In the context of a multiprocessor system, a common initiator of a single memory read or write or MB operation will not be a user, as is stated in that paragraph where the applicant refers to tasks. Nor will the Operating System initiate such low level activities when it initiates a "housekeeping function." Instead, single program statements of the housekeeping function when processed by a processor having that housekeeping task may perform, if at all, such operations similar to the ones referred to at this citation to column 4 of Steely. These are not "tasks" as the claim uses that term. In fact, an electronic search of the content of Steely on the USPTO website shows that Steely never uses the word task at all. The level of invention within a computer system the applicant has described is not dealing with issues of arbitration and cache coherency for memory operations. Steely's disclosure would appear to be addressing that level and type of problem within a computer. While Steely may be interesting and the examiner may be trying to say that the concepts of Steely can apply at this level to OS task assignment, the examiner has not said so, instead saying that Steely is doing the same thing as the applicant has claimed. But Steely's cited to activities are not the same, and the citations to Steely do not support such a

statement. The applicant is teaching how to apportion processor time to accomplish multi-step tasks among multiple processors while Steely appears to be trying to prevent screw-ups in consistency and coherency among memory operations using a careful application of a Memory Barrier operation in pipelined activity for individual ones of multiple processors. The goals are not the same and the systems do not share the features the examiner claims they do. As pointed out above, Steely has a simpler memory hierarchy that does not operate in the same way that the applicant's claimed hierarchy operates. Because there are no tasks of the kind being managed by the applicant's invention in Steely, there are no switching queues of the type described for use to manage such task types as are in the applicant's invention either. The memory hierarchy for the applicant provides an opportunity to assign tasks to groups of processors sharing various levels of cache memory structures at the various levels of memory to get more "hits" on memory in the various levels of caches, by preferentially assigning tasks to processors sharing caches at various levels. Thus tasks will run more efficiently within the memory architecture described by the applicant for this invention than running them without the applicant's invention. There is no opportunity to take advantage of this in Steely. This opportunity is capitalized upon by the applicant's invention.

This assignment of affinity is to the task information or header in preferred embodiments. (Page 6 last full paragraph and page 7 first full paragraph, lines 7 and 8 of claim 1). There is no corresponding way to assign SQs to MB operations in Steely. Thus Steely fails to show this limitation also as there is no control data structure for each task affined to a processor SQ in Steely, because there is no task, there is no affinitization, and there is no control data structure (such as the task header just mentioned into which the assignment can be made).

Accordingly, as Steely fails to show many of the limitations of the claims, reconsideration and withdrawal of the rejection is respectfully solicited.

The Gove reference appears to be similarly inapposite, being directed to massively parallel systems and not addressing operating systems for multiprocessors with more than two levels of hierarchical memory. In any event it does not supply the missing elements.

Claims 18-25 are rejected over Steely and Gove as well. This rejection is deficient in the same ways as the rejection based on these references discussed above. Further, this rejection states that Steely teaches a dispatcher algorithm, which is also incorrect. The applicant uses the term dispatcher as a component of an Operating System. The cited to column and lines for this term do not show such a component in Steely.

With respect, the official communication fails to discover components in the references that match the claims limitations, and no a priori case is made out. Accordingly, the rejection is traversed.

Reconsideration and withdrawal is respectfully requested and further examination or allowance of all claims is respectfully solicited.

Respectfully submitted,

February 24, 2005

Michael B. Atlass
Attorney for Applicants
Registration No. 30,606

UNISYS CORPORATION
Unisys Way, MS/E8-114
Blue Bell, Pennsylvania 19424-0001
Ph.: (215) 986-4111
Fax: (215) 986-3090